

# **Appendix 1**

## **to Tender Specifications**

# **Project Delivery**

---

## Table of Contents

1.	Project Phases.....	4
1.1.	Initiation.....	4
1.2.	Design.....	4
1.3.	IMPLEMENTATION.....	5
1.3.1.	Factory Acceptance Test (FAT) .....	5
1.3.1.1.	Test Readiness Review .....	5
1.3.1.2.	FAT Execution .....	6
1.4.	EMSA TEST and Deployment .....	6
1.5.	Training.....	6
1.6.	Go-Live .....	6
2.	PROJECT DELIVERABLES .....	7
2.1.	Initiation.....	7
2.1.1	Project Management documentation.....	7
2.1.2	Software Development Plan .....	7
2.1.3	Tools for project controlling and design .....	8
2.2.	Design.....	8
2.2.1	Design Documentation .....	8
2.2.1.1.	Functional Design Specifications .....	8
2.2.1.2.	Technical Design Specifications.....	8
2.2.1.3.	Draft version of the Software Test Plan.....	9
2.3.	Development and Test.....	9
2.3.1	System documentation.....	9
2.3.1.1.	Operational and Maintenance Documentation .....	9
2.3.1.2.	System building procedures .....	10
2.3.1.3.	Infrastructure (HW and SW) documentation .....	10
2.3.2	A Version of the system.....	10
2.3.3	Source code management.....	11
2.3.4	Build management.....	11
2.3.5	Unit and integration tests .....	11
2.3.6	Deployment automation .....	11
2.3.7	User Documentation.....	12
2.3.1.4.	User documentation .....	12
2.3.1.5.	Training support material.....	12
2.3.8	Test Documentation .....	12
2.3.9	Test Automation .....	13

---

2.4.	Deployment and DATA IMPORT.....	13
2.5.	Go-Live .....	14
2.6.	Training.....	14
3.	Acceptance procedures.....	14
3.1.	Classification of SOFTWARE issues .....	14
3.2.	Documentation .....	15
3.3.	version of the system .....	15
3.4.	Final system.....	16
4.	Meetings.....	16
4.1.	Project management meetings .....	16
5.	ENVIRONMENTS .....	16

This document details the requirements of the information system for the provision of services until final acceptance of the system excluding warranty.

Phase 1 of the project is also composed by specific phases. This document presents all these specific phases of Phase 1. It describes what deliverables are requested, when they should be delivered and how they will be accepted by EMSA.

## **1. PROJECT PHASES**

Six major phases are foreseen from signature of the contract until the final system is accepted, which are:

- Initiation
- Design
- Development and Test
- Deployment
- Go-Live
- Training

Deliverables are expected for each phase. They are detailed in Chapter 2 – Project Deliverables of this document.

If one deliverable is not accepted by EMSA during one phase no acceptance of any deliverable of the following phase can be done.

### **1.1. INITIATION**

The objective of this initiation phase is to have a mutual understanding and agreement of methods and means that will be used for the completion of the project.

Immediately after the signature of the contract the contractor should prepare the kick-off meeting to cover at least the following subjects:

- objectives and organization
- contractor team
- project tools
- project plan
- methodologies and procedures
- software development plan
- content and level of detail of the project management documentation

During this phase, the contractor is asked to work in close contact with EMSA in order to create a common view of the whole project.

### **1.2. DESIGN**

The objective of the design phase is to create a complete set of functional and technical specifications specifying what and how is to be implemented and the methodologies that shall be used to verify and validate the project execution.

During this phase the contractor shall deliver the following documents for review of the design and approval by EMSA:

1. Technical Design Specification: main design document of the application, updated with the changes implementing the release requirements
2. Interface Control Document: specifying the application external interfaces. The delivered version shall highlight the changes introduced in the new version.

### 1.3. IMPLEMENTATION

The objective of this phase is to develop and test the final version of the system before deployment and import of data.

The application will be developed according to the deliverables of the design phase.

Before delivery the contractor must test the developed application to verify the conformity with expected results and validate that the procedures as stated during the previous phases have been applied. The contractor should respect the "two chamber principle" which means that the team in charge of the tests should be different from the team in charge of the design and development. **Tests cannot be executed prior acceptance of Test documentation by EMSA.** Test report and test evidence should be transmitted to EMSA.

During this phase the contractor is responsible to:

- Prepare system documentation;
- Deliver a final version of the system;
- Prepare User documentation, including training support material;
- Prepare Test documentation;
- Test final version before its delivery and report test results to EMSA.

#### 1.3.1. FACTORY ACCEPTANCE TEST (FAT)

The development phase concludes with the Factory Acceptance Test. The purpose of this milestone is to validate that the release is ready for delivery. All requested functionality has been implemented and tested.

##### 1.3.1.1. TEST READINESS REVIEW

The contractor delivers the following documents for review at least 10 working days prior to the execution of the FAT. The test readiness review aims at checking that the FAT milestone can be held.

The delivery shall include at least:

1. Test plan, describing how the delivery requirements will be validated
2. Interface Control document (final) including all updates
3. Test report and Test evidence for the tests to be run in the FAT milestone. FAT tests missing at this stage shall be listed
4. The set of application documents impacted by the release. EMSA shall provide feedback on the bundle within three working days. FAT may be rescheduled if the required coverage of the requirements is not met.

EMSA will decide if the FAT milestone can be held on the basis of these deliveries and the development progress tracking described in 2.3.

#### **1.3.1.2. FAT EXECUTION**

The FAT is executed in a representative environment provided by the contractor, including all necessary test tools and infrastructure to validate the release requirements.

The FAT shall include all tests necessary to validate both functional and non-functional (e.g. performance and availability) requirements, unless otherwise agreed with EMSA.

The duration of the FAT for an EMSA application release is typically 2 or 3 days, but it might be longer or shorter, depending on the scope.

Prior to the FAT, the contractor should provide report and evidence of at least one long running test (minimum 24 hours) with representative load, in order to demonstrate the system stability.

The outcome of the milestone is based on the FAT test results demonstrating the compliance of the application release to the requirements, to the level specified.

#### **1.4. EMSA TEST AND DEPLOYMENT**

Immediately after the validation and delivery of a release, EMSA will start its deployment.

The objective of the deployment phase is to configure and make the final version available and fully running on its environments:

- Test;
- Pre-Production/Quality;
- Training (for applications having a training environment)
- BCF
- Production.

During this phase EMSA will perform acceptance tests to accept the final version, the system documentation, the user documentation and the training materials.

#### **1.5. TRAINING**

The training phase starts after acceptance of final version, user documentation and training materials.

During this phase the contractor should organise and conduct the following training session:

- teach EMSA's operational unit end-users to use the application,
- teach EMSA's IT personnel to manage and administrate the system.

Training will be done on the test and/or quality environment.

In case adjustments or corrections are found necessary during training sessions the contractor will be asked to update user documentation and training materials

#### **1.6. GO-LIVE**

The go-live phase starts after the final version is accepted by EMSA. The objective of this phase is to:

- obtain an optimum configuration of the system and maximal performance in the production environment by fine tuning the complete technical infrastructure,

- perform necessary correction and adjustments of the system while it is used by end users in real situation

The go-live phase ends at the final acceptance of the system.

## 2. PROJECT DELIVERABLES

Documentation must be provided in electronic format compatible with MS Office 2010.

### 2.1. INITIATION

#### 2.1.1 Project Management documentation

Project management documentation should reflect the project management methodology proposed by the contractor in its bid. It should include at least the following documents:

- **Project plan:** must include the following items at least: project charter, project management approach, scope, Work Breakdown Structure (WBS), project team, Gantt chart, deliverables milestones, working locations, meetings planning and reports.
- **List of outstanding and closed Action Items.**
- **Highlight report:** weekly report on the status on the project containing (at least) ongoing tasks, resources usage, progress status, and issues foreseen.
- **Agenda of the meetings:** the contractor is responsible for providing detailed agenda and additional requests 3 days before the meetings for all relevant meetings held between EMSA and the contractor.
- **Minutes of the meetings:** the contractor is responsible for providing the minutes of the meetings for all relevant meetings held between EMSA and the contractor. The minutes of the meetings must include at least the topics discussed, decisions taken and action items with indication of the responsible person and deadline of the actions.

#### 2.1.2 Software Development Plan

The Software Development Plan identifies the general methodologies, processes and working practices to be used on the development of the project. It serves as the basic rules and practices guideline for all the remaining technical tasks of the project. The Software Development Plan will at least address the following issues:

- Software development approach: description of the strategy of the software development life cycle (waterfall, incremental, evolutionary life cycle, etc)
- Software engineering environment: identification of any standards, methods and tools that apply to the project, whether defined by the contractor or required by EMSA. Details regarding the application of each item identified shall be addressed on the subsequent sections when referring them.
- Software quality assurance process: definition of rules, practices and conventions to be used as to obtain the desired quality of the final product and to evaluate if this plan is being properly implemented. References to standards or one or more contractor's own documents are possible.
- Software configuration management plan: definition of rules, practices, conventions and tools to be used as to maintain the software configuration. References to standards or one or more contractor's own documents are possible.

- Design standards: definition of rules, practices, conventions and tools to be used in definition of the design. References to standards or one or more contactor's own documents are possible.
- Coding standards: definition of rules, practices, conventions and tools to be used in development of the code. References to standards or one or more contactor's own documents are possible.
- Testing standards and practices: identifies the standards, practices, conventions and tools that will be used to test the developed software. References to standards or one or more contactor's own documents are possible.

### 2.1.3 Tools for project controlling and design

EMSA teamforge is the mandatory tool for project issue tracking and requirements tracking.

Unified Modelling Language (UML) should be used for object and system modelling. EMSA suggests the use of Enterprise Architect UML tool. The contractor is free to use another UML modelling tool as long as he guarantees its compatibility with one of the previous tools, however prior agreement with EMSA.

Project progress controlling shall be EVM (Earned Value Management).

Development progress shall be tracked using automated reporting based on the process described in 2.2.

## 2.2. DESIGN

Design documentation should be prepared in close collaboration with EMSA's personnel.

### 2.2.1 Design Documentation

Design Documentation should cover:

1. Functional design specifications,
2. Technical design specifications,
3. A draft version of the Software Test Plan containing at least the test strategy.

#### 2.2.1.1. Functional Design Specifications

Functional design specifications will be used as guidelines for the implementation of the system.

They should describe in detail as a minimum of:

- Use Cases representing the system functionalities,
- capabilities and processes,
- interactions with users/systems,
- Traceability matrix between Functional Requirements/Business Rules and Use Cases.

#### 2.2.1.2. Technical Design Specifications

Technical design specifications will be used as a blueprint for the system implementation. They describe how the system will be implemented in order to cope with functional specifications.

They should include as a minimum:

- Conceptual and physical system architecture,
- Software design and layering,
- Modules and components,
- Process, workflows and algorithms design and documentation,
- Interfaces definitions.

#### 2.2.1.3. Draft version of the Software Test Plan

The draft version of the Software Test Plan will serve as the basis for the Software Test Plan to be implemented during “Development and Test” phase. This draft version should include as a minimum:

- Definition of the Software Test Plan Structure and global strategy,
- Reference to the different test phases to be implemented,
- Definition of the test detailed strategy presenting an overall perspective of testing and identifying individual test phase plans for unit, integration, functional, performance, load and stress test phases. Each test phase plan should include at least:
  - Description of the test phase strategy,
  - Test phase standards and practices,
  - Test phase supporting guidelines,
  - Test phase selection criteria,
  - Test phase evaluation metrics,
  - Completion criteria for the test phase,
  - Test phase implementation templates,
  - Requirements Traceability Verification (ensuring that each requirement has one or more test cases associated with it)
- Reference to the test environment(s) to be used,
- Software Test plan execution planning,
- Software Test team responsibilities and staff.

The final version of the Software Test Plan is to be provided during the Development and Test phase.

## 2.3. DEVELOPMENT AND TEST

If needed the contractor and/or EMSA may suggest modifying the content of the deliverables of the design phase. These modifications should be agreed by EMSA.

### 2.3.1 System documentation

#### 2.3.1.1. Operational and Maintenance Documentation

Operational and maintenance documentation must explain how the system should be operated and maintained on a daily base. It should include the following documentation:

- Installation manual.
- Operation and Maintenance Manual.
- Incident Handling Procedures. HOW-TO troubleshooting and root-cause analysis.

#### 2.3.1.2. System building procedures

System building procedures should allow EMSA to completely build the latest version of the system at any moment.

System building procedures shall be executed in EMSA building environment. The contractor has to provide all the needed information to EMSA to prepare this building environment. EMSA favours the use of virtual Linux building environments.

At the delivery of the final version the contractor should provide an automatic build procedure with the complete source code, additional software packages and code generators.

For each code generator used during development a corresponding generator should be provided to EMSA.

#### 2.3.1.3. Infrastructure (HW and SW) documentation

The contractor is requested to provide a complete and detailed architecture definition and sizing for the following environment:

- Test;
- Pre-Production/Quality;
- Training;
- BCF;
- Production.

The environments will be provided at EMSA Data Centre.

In order to correctly size the production environment the contractor must consider the following elements: system architecture, implementation, non-functional requirements and the performance requirements specified in Annex I – Tender Specifications.

For the production environment, detailed information about requirements for servers characteristics, network, bandwidth, base software, databases, security and accessibility shall be provided to EMSA. For the others environments, the same level of information must be provided with an indication of expected performance.

### 2.3.2 A Version of the system

A version of the system delivered to EMSA should contain:

- System implementation
- Related source codes, build procedures and supporting documentation,
- A complete system documentation,
- Test documentation including automated tests for at least all non GUI tests and performance tests using EMSA approved test tools (JMeter, SoapUI, ReadyAPI SoapUI Pro, Robot Framework),
- Automated deployment scripts (puppet)
- Release notes.

---

### 2.3.3 Source code management

During development, the contractor shall use EMSA's GitLab repository as their main source code repository and make daily commits to it. Rather than delivering source code to EMSA, at the time of release, EMSA will merge the development branch with the main branch when the contractor makes a formal delivery. EMSA will also perform a daily build of the code in the development branch, run automated tests and perform automated quality checks of the code as described in the EMSA quality gate (see appendix 2).

### 2.3.4 Build management

EMSA currently uses the following applications in building and testing the source code delivered by contractors: Maven, Jenkins, Sonar and Archiva.

Hudson checks out the source code from GitLab (either latest version or a specific tag) and builds it locally. The build definition is contained in a Maven POM. After a successful build, the CI server will trigger the unit tests, build the test coverage report and trigger Sonar to run the code quality measurements. It will also upload the generated build artifacts to Archiva. The CI server also provides a dashboard with the overview of the status of all projects. Contractor shall deliver Maven POMs for the application.

During development a daily build cycle in EMSA will be put in place. Jenkins will be exposed outside, allowing contractors to verify results of the build process and upload new components, i.e. the contractor will have read access only to his own projects. EMSA may trigger an alert/veto on upload of new components. Builds will automatically run every night; and results will be made available to EMSA through a dashboard for progress tracking purposes.

### 2.3.5 Unit and integration tests

The contractors shall deliver unit tests alongside the source code to be run after the daily build. Unit test coverage requirements are specified in Appendix 2 (EMSA Quality Gate)

Unit tests will be run every day by the CI tools based on the daily push. A report containing the results will be sent to all project managers and heads of units.

For every requirement concerning analysis of data, a realistic set of data has to be provided in such a way that it can be played back at any time.

### 2.3.6 Deployment automation

EMSA uses puppet as a tool for deployment automation. The contractor shall deliver puppet modules for installation of each release, complemented with other scripting tools (e.g. WLST for Weblogic) where needed. Installation based on unpacking archive files or RPM installation packages should be avoided. The deployment scripts shall not include the installation of middleware and infrastructure components.

Applications shall always be installed from scratch, with the exception of databases for which the contractor shall deliver alongside each version scripts to create from scratch and to update the the new version from the preceding one.

Environment dependent configuration shall be provided to puppet as parameters defined in the Puppet module and therefore will only be specified in the puppet manifest applied during 'puppet apply' or by 'puppet agent'. To avoid unnecessary releases of new puppet modules, version dependencies should also be defined in the same manner.

Administration accounts (root, sysdba,...) usage is not allowed (exceptions can be evaluated)

For tests see chapter "Test Automation" below.

---

## 2.3.7 User Documentation

### 2.3.1.4. User documentation

The user documentation will explain the different components of the system to EMSA's users.

User documentation should include:

- A quick start guide to explain how to access the system and use main functionalities,
- A complete user manual to describe how to use all functionalities of the system,
- On line help content. This will contain a contextual help explaining content and functionalities of every screen of the system and generic help which will provide the same content of the user manual.

Text should be supported by illustrations and screen copies all through the user documentation. User manual and on line help should include a table of content, a glossary and an index.

### 2.3.1.5. Training support material

As a minimum training support material should contain:

- training presentations to be displayed during the sessions,
- practical examples,
- training data sets – if applicable.

## 2.3.8 Test Documentation

Tests to be performed by the contractor must cover the two following objectives:

- Verification tests: verify that the product is in line with the functional and technical requirements and design specifications and that implementation best practices were applied,
- Validation tests: verify that procedures and activities as described in the project plan, change management procedures and software development plan were applied.

The test documentation and test results should provide evidence that these objectives are met.

Test documentation should detail all necessary documents to plan, design, execute and report tests. This should include as a minimum:

- The Software Test Plan with all details regarding the test process:
  - Definition of the Software Test Plan Structure and global strategy,
  - Reference to the different test phases to be implemented,
  - Definition of the test detailed strategy presenting an overall perspective of testing and identifying individual test phase plans for unit, integration, functional, performance, load and stress test phases. Each test phase plan should include at least:
    - Description of the test phase strategy,
    - Test phase standards and practices,
    - Test phase supporting guidelines,
    - Test phase selection criteria,
    - Test phase evaluation metrics,

- Completion criteria for the test phase,
- Test phase implementation templates.
- Results achieved with the test phase implementation including at least:
  - Test cases,
  - Test scripts,
  - Data sets,
  - Test results,
  - Test phase report.
- Reference to the test environment(s) to be used,
- Software Test plan execution planning,
- Software Test team responsibilities and staff.

Test results should be added to each test plan once the corresponding tests have been executed.

The contractor will be responsible for preparing all documentation including test cases, test data to be used and test environment.

### 2.3.9 Test Automation

Test automation scripts shall be provided (and maintained) within the scope of the project as well a consistent and meaningful Test Dataset.

- Unit Test (See Appendix 2)
- Test Scripts
  - Initially, must cover meaningful Test Cases for a set of critical business functions (to be agreed) to always be executed as regression tests
  - Test Scripts must be enriched with at each delivery:
    - New Test Cases for critical business functions either new or not yet covered
    - New Test Cases to allow the specific validation and verification (V&V) of critical and blocking error found in the previous versions
- Consistent Test Dataset to be provided (and maintained)
  - Test Dataset must be automatically loaded into the TEST environment (using a data loading mechanism)
  - Initial dataset must be aligned with the Test Scripts and Test Cases agreed
  - Test Dataset must be enriched with:
    - Data for new Test Cases
    - New data to allow V&V of critical and blocking error found in the previous versions
- SOAPUI and JMeter shall be used for test automation to the extent possible.

## 2.4. DEPLOYMENT AND DATA IMPORT

Deliverables of the deployment phase are:

- A version of the system deployed and fully working in the three environments (or 5 in case of training and BCF).
- Updates of the system documentation, user documentation and training materials if needed.
- Updates of the deliverables of the design phase if needed.

## **2.5. GO-LIVE**

Deliverables of the Go-live phase are:

- Updates of the system documentation if needed,
- Report on the tasks undertaken by the contractor and their results,
- Final system.

## **2.6. TRAINING**

Training sessions are foreseen for:

- EMSA's operational unit end-users,
- EMSA's IT personnel.

All sessions should mix theoretical and practical parts. In the practical parts, the users actually use the system with hands on examples.

The contractor will be responsible to conduct training sessions. The contractor should provide all supporting material and prepare hands on examples. It will also be responsible for the preparation of the technical environment (software and data).

Training sessions will take place at EMSA's premises in Lisbon. EMSA will provide infrastructure (rooms, IT equipment, video equipment etc.) and will be in charge of administrative organisation of the courses (planning, notifications, and evaluation) and duplication and distribution of course documentation.

## **3. ACCEPTANCE PROCEDURES**

For each deliverable, EMSA provides a formal indication of the acceptance, conditional acceptance or rejection of the deliverable to the contractor.

### **3.1. CLASSIFICATION OF SOFTWARE ISSUES**

EMSA will classify issues found on software into 3 different categories according to their impact and severity as described in Annex VII of the Framework Contract.

The outcome of the acceptance procedure is positive if no issue is found by EMSA. If issues are found by EMSA during the acceptance procedure, the contractor is requested to immediately correct them and the acceptance procedure restarts from the date of the delivery of the corrected deliverable.

EMSA can decide to conditionally accept the deliverable when some issues remain uncorrected if those issues are not blocking the usage of system. In order to accept such remaining issues the contractor shall propose a deadline for the correction and EMSA to accept it. The EMSA will take the decision on conditionally acceptance of the product after evaluation of each remaining issue.

EMSA reserves the right to perform code reviews, either during the project execution or before final acceptance. During a code review, attention will be given to the topics described in Chapter 17 "Code smells and Heuristics" of the "Clean code" book<sup>1</sup>. Based on this review, if a large number of issues are discovered, the delivery may not be accepted until the issues have been solved.

No acceptance shall be made by EMSA without a successful execution of the automatic build procedure nor without successfully installing all deliveries in EMSA quality environment.

### 3.2. DOCUMENTATION

In the case of Project Management documents, EMSA will provide comment and/or reservations which will be transmitted to the contractor within **one week** of the date of delivery. Based on this comment and/or reservations EMSA will either accept or reject the deliverables. In the case of rejection the contractor will be requested to provide a new appropriate revision.

In the case System documentation and User Documentation, EMSA will provide comment and/or reservations which will be transmitted to the contractor within **two weeks** of the date of delivery. Based on this comment and/or reservations EMSA will either accept or reject the deliverables. In the case of rejection the contractor will be requested to provide a new appropriate revision.

In the case of Design Documentation, EMSA will provide comment and/or reservations which will be transmitted to the contractor within **five days** of the date of delivery. Based on this comment and/or reservations EMSA will either accept or reject the deliverables. In the case of rejection the contractor will be requested to provide a new appropriate revision.

### 3.3. VERSION OF THE SYSTEM

A version of the system will be evaluated by EMSA when available and running on the test, quality, production and BCF environments.

Before the final version is accepted EMSA will verify if:

- all issues detected in the previous acceptance tests have been corrected,
- it conforms with the functional specifications,
- it conforms with the technical specifications,
- non-functional requirements are met,
- it works correctly in its environments according to all requirements and specifications.

EMSA will provide issues which will be transmitted to the contractor within **fourteen days** of the date of delivery. Based on this issues EMSA will either accept or reject the version. In the case of rejection the contractor will be requested to provide a new appropriate version.

---

<sup>1</sup> Clean Code by Robert C. Martin

Publisher: Prentice Hall

Release Date: August 2008

ISBN: 9780136083238

### 3.4. FINAL SYSTEM

The final system will be evaluated by EMSA when the accepted final version will be available in the production environment. EMSA will verify the system operates correctly while being used by end users in real situation.

The Final system is accepted within acceptance period (**14 days**) at the condition that no urgent issues as described in chapter 3.1 are found.

In the case a blocking issue is found, the acceptance period is reset until a corrected version is made available on the production environment by the contractor.

## 4. MEETINGS

### 4.1. PROJECT MANAGEMENT MEETINGS

Action list, risk registry and planning will be reviewed during project management meetings.

At each project management meeting, the contractor should present an updated project status report.

In addition to the project status reports, between the project management meetings, the contractor delivers to EMSA a flash report.

The contractor is responsible for providing detailed agenda and supporting documents for the meetings, support the discussions during the meeting, and providing the minutes of the meetings. The detailed agenda and supporting documents must be provided by the contractor 3 days before each meetings. The minutes of the meetings must include at least the topics discussed, decisions taken and action items with indication of the responsible person and deadline of the actions.

## 5. ENVIRONMENTS

### Development environment is the Contractor's responsibility

- The Contractor is responsible for setting up and maintaining the development environments and their own integration test environments if needed.

### TEST Environment

- Managed by EMSA ICT but in a relaxed way
- The Contractor has access to TEST Environment without root (SO) or sysdba (Database) privileges

### PRE-PRODUCTION Environment

- Managed by EMSA ICT
- Open to internet through IP filtering
- Mimics PRODUCTION as much as possible: same architecture, same versions
- The Contractor has limited access to PRE-PROD environment

### TRAINING Environment

- Managed by EMSA ICT
- Open to internet
- Smaller infrastructure with reduced architecture with the same versions
- The Contractor has limited access to Training environment

**PRODUCTION and BCF Environments**

- Fully managed and controlled by EMSA ICT
- Monitored 24x7
- The Contractor will not have access to PRODUCTION (exceptions might be considered case by case)

Each environment shall have its own set of application configuration files.

Application configuration files shall be externalized from the application binaries

[End of document]